I. Amendments to the Claims

This listing of claims replaces without prejudice all prior versions and listings of claims in the application:

Listing of Claims:

1. (Original) In a pointer tracking system including at least two overlapping coordinate input sub-regions defining a generally contiguous input region, each coordinate input sub-region generating pointer coordinate data in response to pointer movement therein, a method for tracking a pointer across overlapping portions of said coordinate input sub-regions comprising:

detecting pointer movements within overlapping portions of said coordinate input sub-regions; and

processing the pointer coordinate data generated by each of said coordinate input sub-regions as a result of pointer movement within said overlapping portions in accordance with defined logic to yield a single set of pointer coordinate data representing the pointer movement.

2. (Original) The method of claim 1 wherein during said processing the pointer coordinate data is combined in accordance with said defined logic.

3. (Original) The method of claim 2 wherein said defined logic is an averaging technique.

4. (Original) The method of claim 3 wherein said coordinate input sub-regions only partially overlap.

5. (Original) The method of claim 4 wherein said averaging technique is a weighted averaging technique.

6. (Original) The method of claim 5 wherein said pointer coordinate data includes a series of pointer (x,y)-coordinates and wherein the pointer coordinate data is combined according to the equation:

y-coordinate = (100 – P%) * y-coordinate of CIR_x + P% * y-coordinate of CIR_{x+1}

where:

CIR_x is one coordinate input sub-region;

CIR_{x+1} is another coordinate input sub-region; and

P% is the distance travelled through the overlapping portions in an x-direction expressed as a percentage when travelling in a direction from coordinate input sub-region CIR_x to coordinate input sub-region CIR_{x+1}.

7. (Original) The method of claim 2 wherein said coordinate input sub-regions only partially overlap and wherein said defined logic is a weighted averaging technique.

8. (Original) The method of claim 7 wherein each coordinate input sub-region generates pointer coordinate data by:

capturing overlapping images looking across the coordinate input sub-region;

detecting the presence of a pointer in each of the captured images; and

triangulating the detected pointers to determine (x,y)-coordinates of said pointer.

9. (Original) The method of claim 8 wherein said pointer coordinate data includes a series of pointer (x,y)-coordinates and wherein the pointer coordinate data is combined according to the equation:

y-coordinate = (100 – P%) * y-coordinate of CIR_x + P% * y-coordinate of CIR_{x+1}

where:

CIR_x is one coordinate input sub-region;

CIR_{x+1} is another coordinate input sub-region; and

P% is the distance travelled through the overlapping portions in an x-direction expressed as a percentage when travelling in a direction from coordinate input sub-region CIR_x to coordinate input sub-region CIR_{x+1}.

10. (Original) The method of claim 1 further comprising displaying an image generally spanning said contiguous input region, said image being updated to reflect pointer activity.

11. (Original) The method of claim 10 wherein said image includes image segments, each segment being associated with a respective coordinate input sub-region.

12. (Original) The method of claim 11 wherein image segments associated with adjacent coordinate input sub-regions are joined substantially seamlessly within said overlapping portions of said coordinate input sub-regions.

13. (Original) The method of claim 11 wherein said image segments are operating system desktop sections.

14. (Original) The method of claim 11 wherein said coordinate input sub-regions only partially overlap and wherein said defined logic is a weighted averaging technique.

15. (Original) The method of claim 14 wherein each coordinate input sub-region generates pointer coordinate data by:

capturing overlapping images looking across the coordinate input sub-region;

detecting the presence of a pointer in each of the captured images; and

triangulating the detected pointers to determine (x,y)-coordinates of said pointer.

16. (Original) The method of claim 15 wherein said pointer coordinate data includes a series of pointer (x,y)-coordinates and wherein the pointer coordinate data is combined according to the equation:

y-coordinate = (100 – P%) * y-coordinate of CIR_x + P% * y-coordinate of CIR_{x+1}

where:

CIR_x is one coordinate input sub-region;

CIR_{x+1} is another coordinate input sub-region; and

P% is the distance travelled through the overlapping portions in an x-direction expressed as a percentage when travelling in a direction from coordinate input sub-region CIR_x to coordinate input sub-region CIR_{x+1}.

17. (Original) In a touch system including a plurality of coordinate input sub-regions that overlap defining a generally contiguous input surface, each coordinate input sub-region generating pointer coordinate data in response to pointer contacts thereon, said pointer coordinate data being processed to update image data presented on said input surface, a method of detecting the position of a pointer contact relative to said touch surface comprising:

acquiring overlapping images of each coordinate input sub-region;

when a pointer contact is made on a portion of a coordinate input sub-region that does not overlap with an adjacent coordinate input sub-region, processing acquired images to derive pointer data and triangulating the position of the pointer using the derived pointer data thereby to determine the position of the pointer contact relative to the touch surface; and

when a pointer contact is made on a portion of a coordinate input sub-region that overlaps with an adjacent coordinate input sub-region, for each coordinate input sub-region processing acquired images to derive pointer data, and triangulating positions of the pointer using the derived pointer data, and thereafter processing the triangulated positions in accordance with defined logic thereby to determine the position of the pointer contact relative to the touch surface.

18. (Original) The method of claim 17 wherein during said processing the triangulated positions are combined.

19. (Original) The method of claim 18 wherein said triangulated positions are combined using weighted averaging.

20. (Original) The method of claim 19 further comprising updating the image data in accordance with the determined position of the pointer contact relative to the touch surface.

21. (Original) The method of claim 20 further comprising maintaining attributes assigned to a pointer by one coordinate input sub-region after said pointer moves across an overlapping portion into an adjacent coordinate input sub-region.

22. (Original) The method of claim 21 wherein said attributes are maintained until a pre-defined event occurs.

23. (Original) The method of claim 19 wherein said coordinate input sub-regions only partially overlap.

24. (Original) The method of claim 23 wherein said pointer coordinate data includes a series of pointer (x,y)-coordinates and wherein the pointer coordinate data is combined according to the equation:

y-coordinate = (100 – P%) * y-coordinate of CIR_x + P% * y-coordinate of CIR_{x+1}

where:

CIR_x is one coordinate input sub-region;

CIR_{x+1} is another coordinate input sub-region; and

P% is the distance travelled through the overlapping portions in an x-direction expressed as a percentage when travelling in a direction from coordinate input sub-region CIR_x to coordinate input sub-region CIR_{x+1}.

25. (Original) The method of claim 19 wherein said coordinate input sub-regions only partially overlap.

26. (Original) The method of claim 25 wherein said image includes image segments, each segment being associated with a respective coordinate input sub-region.

27. (Original) The method of claim 26 wherein image segments associated with adjacent coordinate input sub-regions are joined substantially seamlessly within overlapping portions of said coordinate input sub-regions.

28. (Original) The method of claim 26 wherein said image segments are operating system desktop sections.

29. (Original) The method of claim 25 further comprising maintaining attributes assigned to a pointer by one coordinate input sub-region after said pointer moves across an overlapping portion into an adjacent coordinate input sub-region.

30. (Original) A touch system comprising:

a plurality of coordinate input sub-regions, said input sub-regions overlapping to define a generally contiguous input surface, each coordinate input sub-region acquiring overlapping images thereof and generating pointer coordinate data in response to pointer contacts thereon, said pointer coordinate data being processed to update image data presented on said input surface, wherein:

when a pointer contact is made on a portion of a coordinate input sub-region that does not overlap with an adjacent coordinate input sub-region, said coordinate input sub-region processes acquired images to derive pointer data and triangulates the position of the pointer using the derived pointer data thereby to determine the position of the pointer contact relative to the touch surface; and

when a pointer contact is made on a portion of a coordinate input sub-region that overlaps with an adjacent coordinate input sub-region, each overlapping coordinate input sub-region processes acquired images to derive pointer data and triangulates the position of the pointer using the derived pointer data, the triangulated positions generated by the overlapping coordinate input sub-regions being processed in accordance with defined logic thereby to determine the position of the pointer contact relative to the touch surface.

31. (Original) A touch system according to claim 30 wherein said coordinate input sub-regions only partially overlap.

32. (Original) A touch system according to claim 31 wherein when a pointer contact is made on a portion of a coordinate input sub-region that overlaps with an adjacent coordinate input sub-region the triangulated positions are combined using weighted averaging.

33. (Original) A touch system according to claim 33 wherein an image segment is presented on each coordinate input sub-region, said image segments being joined to create a continuous image on said input surface.

34. (Original) A touch system according to claim 33 wherein each image section is an operating system desktop section.

35. (Original) A touch system according to claim 33 wherein each coordinate input sub-region includes at least two cameras to capture overlapping images thereof.

36. (Original) A touch system according to claim 35 wherein each coordinate input sub-region includes four cameras.

37. (Original) A touch system according to claim 36 wherein said coordinate input sub-regions are rectangular, said cameras being positioned at the corners thereof, pairs of said cameras being responsible for acquiring overlapping images of quadrants of said coordinate input sub-regions.

38. (New) A touch system comprising:

a large-scale touch surface;

at least three imaging devices positioned along at least one side of said touch surface at spaced locations, each of said imaging devices looking across at least a

portion of said touch surface, fields of view of said imaging devices overlapping in a manner so that each location on the touch surface falls within the fields of view of at least two imaging devices; and

processing structure communicating with said imaging devices, said processing structure processing image data generated by selected imaging devices capturing images of a pointer contacting said touch surface to calculate the position of the pointer contact using triangulation.

39. (New) A touch system according to claim 38 wherein said processing structure selects imaging devices based on proximity of the pointer to said imaging devices.

40. (New) A touch system according to claim 39 wherein the fields of view of said imaging devices overlap in a manner so that many locations on the touch surface fall within the fields of view of at least three imaging devices.

41. (New) A touch system according to claim 40 wherein said processing structure averages triangulation results when the pointer contact is within the fields of view of at least three imaging devices.

42. (New) A touch system according to claim 40 wherein said imaging devices are laterally spaced along one said touch surface.

43. (New) A touch system according to claim 42 wherein said imaging devices are disposed above the plane of said touch surface.